

Zadanie 1 — typy danych (zamknięte)

Zaznacz poprawny typ danych dla poniższych wartości (jedna odpowiedź poprawna).

a) "123"

- ☐ int
- ☐ float
- ☐ str
- ☐ bool

b) [1, 2, 3]

- ☐ tuple
- ☐ list
- ☐ set
- ☐ dict

c) (True, 2, "a")

- ☐ tuple
- ☐ list
- ☐ dict
- ☐ bool

Zadanie 2 — podstawowa składnia (otwarte)

Podaj wynik wykonania kodu:

```
x = 3
x = x + 2 * 4
print(x)
```

Zadanie 3 — operacje na stringach (otwarte)

Co wypisze poniższy program?

```
s = "python"
print(s[1:4])
print(len(s))
```

Zadanie 4 — listy (otwarte)

Podaj, co będzie wynikiem poniższych operacji:

```
lst = [1, 2, 3]
lst.append(4)
lst[1] = 10
print(lst)
```

Zadanie 5 — tuple (zamknięte)

Które zdanie o tuple (krotkach) w Pythonie jest prawdziwe?

- ☐ Można zmieniać wartości elementów w tuple.
 - ☐ Tuple są szybsze od list i niemodyfikowalne.
 - ☐ Tuple mogą zawierać tylko liczby.
 - ☐ Tuple nie mogą być elementami list.
-

Zadanie 6 — słowniki (otwarte)

Co wypisze poniższy kod?

```
d = {"a": 1, "b": 2}
d["c"] = d["a"] + d["b"]
print(d)
```

Zadanie 7 — funkcje (otwarte)

Co wypisze poniższy program?

```
def add(a, b=5):  
    return a + b  
  
print(add(3))  
print(add(3, 2))
```

Zadanie 8 — błędy składni (otwarte)

Wskaż błąd w poniższym fragmencie i popraw go:

```
if x > 10  
    print("OK")
```

Zadanie 9 — pisanie kodu (otwarte)

Napisz funkcję `first_and_last(lst: list) -> tuple`, która:

- przyjmuje listę `lst`,
- zwraca tuple zawierającą pierwszy i ostatni element listy.

Przykład:

```
first_and_last([10, 20, 30]) # powinno zwrócić (10, 30)
```

Zadanie 10 — rozumienie działania kodu (otwarte)

Podaj wynik działania:

```
data = {"imie": "Anna", "wiek": 20}  
keys = list(data.keys())  
values = tuple(data.values())  
print(keys)  
print(values)
```